

REMARKS

Initially, Applicant notes that the remarks and amendments made in this paper are consistent with those presented to the Examiner during the telephone call of April 25, 2008. Accordingly, entry of this amendment and reconsideration of the pending claims is respectfully requested.

By this paper, claims 1, 3, 6, 10, 16-23, 28, 37-38, and 40 have been amended, claim 39 has been canceled, and no claims have been added, such that claims 1-38 and 40 remain pending, of which claims 1, 28, 37, and 40 are the only independent claims at issue.

The Office Action, mailed March 27, 2008, considered and rejected claims 1-40. Claims 1-23 and 27-40 were rejected under 35 U.S.C. § 102(e) as being anticipated by *Schuba* (U.S. Publ. No. 2004/0177139). Claims 24-26 were rejected under 35 U.S.C. § 103(a) as being unpatentable over *Schuba* and *Kuznetsov* (U.S. Publ. No. 2006/0265689).¹ Additionally, claim 28 was objected to for a minor typographical error which is corrected by this amendment.

The presently claimed embodiments are directed to a deterministic rule-based dispatch of objects to code. Claim 1, for example, recites a method for performing rule-based dispatch of a data structure to a group of software methods for further processing. The dispatch is deterministic despite the existence of multiple conflicting rules regarding the dispatch of the data structure. In the method a data structure that is to be processed by a group of one or more software methods is accessed and, subsequent to the act of accessing the data structure, a list of rules is accessed to identify a plurality of rules that apply to the dispatch of the data structure with each of the plurality of rules specifying a different group of one or more software methods to which the data structure should be dispatched. After the evaluating the list of rules, the plurality of rules are resolved to identify only a single prevailing rule that will be applied for the dispatch of the data structure to the group of software methods. The data structure is then dispatched to only the group of one or more methods specified by the single prevailing rule.

The remaining independent claims are closely related to claim 1. Claim 28 recites a storage media storing instructions for implementing the method of claim 1, claims 37 recites a method similar to claim 1, but using functional (i.e., step for) language rather than the specific acts of claim 1 and is narrower in scope, while claim 40 recites a system generally capable of executing the method of claim 1.

¹ Although the prior art status of the cited art is not being challenged at this time, Applicant reserves the right to challenge the prior art status of the cited art at any appropriate time, should it arise. Accordingly, any arguments and amendments made herein should not be construed as acquiescing to any prior art status of the cited art.

While the cited art generally relates to rule-based dispatching of messages, Applicant respectfully submits that the cited art of fails to teach or reasonably support all of the claim elements present in the amended claims. For instance, the cited art fails to teach or support that the methods are software methods, that only a single rule is derived, and that the identification of the single rule occurs at the time specified in the claims. For at least these reasons, Applicant respectfully submits that the presently amended claims are allowable over the cited art.

More particularly, and as reflected above, the presently amended claims recite that the rules are responsible for dispatching the data structure to a group of software methods. In the cited art of Schuba, however, the rules are responsible for dispatching an incoming message to an appropriate network service. For instance, in the background of Schuba it states, "the present invention provides a system that resolves conflicts between rules for network services." The rules of Schuba do not correspond to the rules of the current claims. A network service is not synonymous with a method and therefore, rules related to services are not the same as rules related to methods.

The claimed elements further recite that the list of rules must be resolved to only a single rule. The Office Action admits that Schuba resolves a plurality of rules, but states that a plurality of rules is the same as multiple instances of a single rule resolving. In response to this contention, Applicant notes that the claims have been amended to clarify that only a single rule resolves. As admitted in the Office Action, Schuba resolves multiple rules and therefore does not disclose this limitation. Furthermore, Applicant respectfully submits that the rejection of many of the dependent claims in the prior Office Action was in clear error, because they clearly recited that only a single rule will issue. For instance, claim 10 recites the application of the second prioritization mechanism guarantees that only one rule will prevail under any circumstances from the plurality of rules. The Office Action cite the prioritization of rules as teaching this element, but the prioritization of rules does not guarantee that only one rule will prevail, as claimed. The Office Action cites Applicant's own use of prioritization as evidence that prioritizing results in a single rule. However, it will be noted that the prioritizing of the rules is not the final claimed action, that a single prevailing rule is identified after the list is prioritized. The cited art does not teach using a single rule, and instead specifically states that the multiple rules will be applied in order of priority.

The claims also recite a specific order of events. Namely that the data structure is received, then a plurality of rules are identified that are applicable to the data structure, and then only a single prevailing rule is identified. The Office Action states that these elements are taught by Schuba, because the rule cruncher of Schuba takes a variety of inputs and resolves the rules in the rule

cruncher accordingly. However, such inputs do not correspond to the data structure being received. If they did, those inputs would be required to be dispatched to the group of software methods. Instead, those inputs are used only to modify the rules. An input modifying the rules is not being dispatched to a group of software methods.

In view of the foregoing, Applicant respectfully submits that the other rejections to the claims are now moot and do not, therefore, need to be addressed individually at this time. It will be appreciated, however, that this should not be construed as Applicant acquiescing to any of the purported teachings or assertions made in the last action regarding the cited art or the pending application, including any official notice. Instead, Applicant reserves the right to challenge any of the purported teachings or assertions made in the last action at any appropriate time in the future, should the need arise. Furthermore, to the extent that the Examiner has relied on any Official Notice, explicitly or implicitly, Applicant specifically requests that the Examiner provide references supporting the teachings officially noticed, as well as the required motivation or suggestion to combine the relied upon notice with the other art of record.

In the event that the Examiner finds remaining impediment to a prompt allowance of this application that may be clarified through a telephone interview, the Examiner is requested to contact the undersigned attorney at (801) 533-9800.

Dated this 26th day of June, 2008.

Respectfully submitted,



RICK D. NYDEGGER
Registration No. 28,651
COLBY C. NUTTALL
Registration No. 58,146
JOHN C. BACCOCH
Registration No. 59,890
Attorneys for Applicant
Customer No. 47973